

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES
PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro



(43) Internationales Veröffentlichungsdatum
31. Mai 2001 (31.05.2001)

PCT

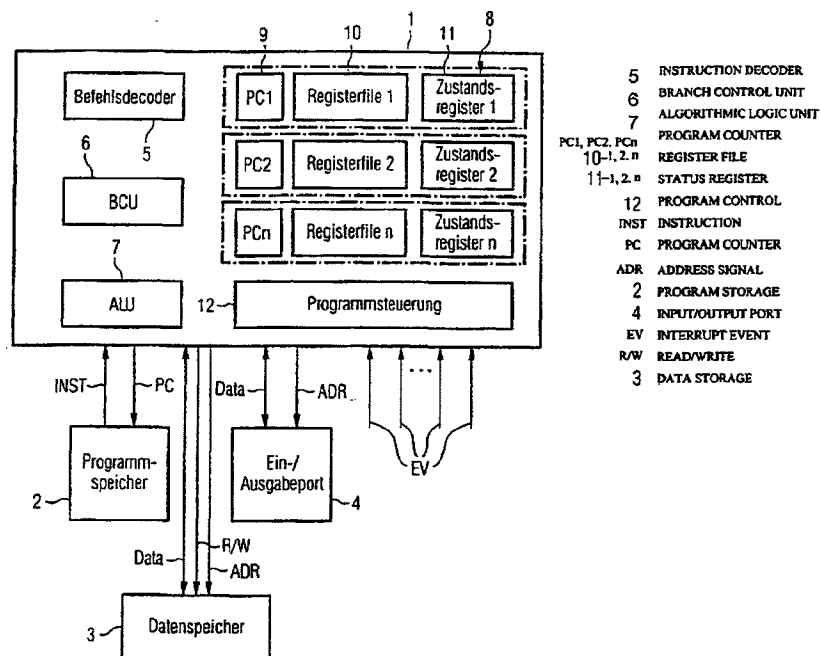
(10) Internationale Veröffentlichungsnummer
WO 01/38968 A2

- (51) Internationale Patentklassifikation⁷: G06F 9/00 (72) Erfinder; und
(75) Erfinder/Anmelder (nur für US): NIE, Xiaoning [DE/DE]; Brehmstrasse 10, 81543 München (DE).
(21) Internationales Aktenzeichen: PCT/EP00/11164 HOBER, Peter [DE/DE]; Sommerstrasse 18, 85521 Riemerling (DE). JENKNER, Christian [AT/AT]; Dr.-Karl-Renner-Weg 12, A-9220 Velden (AT).
(22) Internationales Anmeldedatum: 10. November 2000 (10.11.2000)
(25) Einreichungssprache: Deutsch (74) Anwalt: BARTH, Stephan; Reinhard, Skuhra, Weise & Partner GbR, Postfach 44 01 51, 80750 München (DE).
(26) Veröffentlichungssprache: Deutsch (81) Bestimmungsstaaten (national): CN, JP, KR, US.
(30) Angaben zur Priorität: 199 55 776.4 19. November 1999 (19.11.1999) DE (84) Bestimmungsstaaten (regional): europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).
(71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme von US): INFINEON TECHNOLOGIES AG [DE/DE]; St.-Martin-Strasse 53, 81541 München (DE). Veröffentlicht:
— Ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts.

[Fortsetzung auf der nächsten Seite]

(54) Title: PROCESSOR SYSTEM

(54) Bezeichnung: PROZESSORSYSTEM



(57) Abstract: Different storage means (8) are provided in a processor system for various tasks and can each comprise a stack storage (9) for the instruction counter, a register (10) for intermediately storing data and, and a status register (11). In the occurrence of an interrupt event (EV), which induces a change-over from a momentary task to a new task, a control (21) causes a switching over from storage means (8), which are assigned to the old task, to the storage means (8), which are assigned to the new task.

[Fortsetzung auf der nächsten Seite]



Zur Erklärung der Zweibuchstaben-Codes, und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

(57) Zusammenfassung: In einem Prozessorsystem sind für verschiedene Programmabläufe unterschiedliche Speichermittel (8) vorgesehen, die jeweils einen Stapelspeicher (9) für den Befehlszähler, ein Register (10) zum Zwischenspeichern von Daten und Zustandsregister (11) umfassen können. Bei Auftreten eines Interrupt-Ereignisses (EV), welches einen Wechsel von einem augenblicklichen Programmablauf zu einem neuen Programmablauf hervorruft, wird durch eine Steuerung (21) von den dem alten Programmablauf zugeordneten Speichermitteln (8) auf die dem neuen Programmablauf zugeordneten Speichermitteln (8) umgeschaltet.

Beschreibung

Prozessorsystem

- 5 Die vorliegende Erfindung betrifft ein Prozessorsystem nach dem Oberbegriff des Anspruches 1, welches zur Verarbeitung von Interrupts in der Lage ist.

10 Telekommunikations-Endgeräte, wie beispielsweise Modems oder sogenannte Linecards, müssen auf bestimmte Software- oder Hardwareereignisse reagieren können. Wird zu Steuerungszwecken in derartigen Telekommunikations-Endgeräten ein Prozessorsystem eingesetzt, wird in dem Prozessorsystem zur Reaktion auf die zuvor beschriebenen Ereignisse
15 gewöhnlich eine Interruptverarbeitung vorgesehen.

Zur Interruptverarbeitung werden die einzelnen Ereignisse, die in dem Prozessorsystem einen Interrupt auslösen können, überwacht. Sobald das Auftreten eines Interrupt-Ereignisses
20 erkannt worden ist, wird ein entsprechendes Interrupt-Anforderungssignal generiert und an eine Prozessoreinheit des Prozessorsystems übermittelt, die daraufhin die Startadresse der entsprechenden Interruptroutine, welche zur Ausführung des Interrupts in dem Prozessorsystem ablaufen soll,
25 ermittelt. Treten gleichzeitig mehrere Interrupt-Ereignisse auf, wird aus diesen Interrupt-Ereignissen lediglich ein bestimmtes Interrupt-Ereignis nach einem vorgegebenen Schema zur Ausführung des entsprechenden Interrupts ausgewählt. Der Wert des Programmzählers ("Programm Counter", PC) des vor dem
30 Interrupt in dem Prozessorsystem ablaufen Programmablaufs wird auf einem Stack- oder Stapelspeicher gesichert, so daß anschließend die Interruptroutine ausgeführt werden kann. Nach Beendigung der Interruptroutine kann durch Zugriff auf den gesicherten Programmzähler wieder mit dem ursprünglichen
35 Programmablauf fortgefahren werden.

Vor der Durchführung der Interruptroutine müssen im allgemeinen nicht nur der augenblickliche Wert des Programmzählers, sondern auch weitere Informationen über den Zustand des Prozessorsystems bzw. des darauf ablaufenden Programmablaufs vor dem Auftreten des Interrupt-Ereignisses gesichert werden. Diese Zustandsinformationen können insbesondere die Inhalte von Registern oder Zero- bzw. Carry-Flags etc. betreffen. Zur Beschleunigung der Interruptverarbeitung werden daher in bekannten Prozessorsystemen sogenannte Schattenregister ("Shadow Register") verwendet, welche Zusatzregister darstellen, die nur im Falle eines Interrupts zur Sicherung der obigen Zustandsinformationen in Erscheinung treten und ansonsten nicht verwendet werden (vgl. z.B. "ARM System Architecture", Steve Furber, Addison-Wesley, 1996). Auf diese Weise können zeitaufwendige Registerspeicherbefehle zum Sichern der beschriebenen Informationen vermieden werden.

Der zuvor erwähnte Ansatz stellt jedoch nur dann eine wirksame Maßnahme dar, wenn in dem Prozessorsystem der Ablauf lediglich eines Programmablaufs ("Task") vorgesehen ist. In sogenannten "Multitasking"-Prozessorsystemen, in denen mehrere Programmabläufe gleichzeitig bzw. quasigleichzeitig ablaufen können, kann dieser Ansatz nicht ohne weiteres angewendet werden, da hier für die einzelnen Programmabläufe unterschiedliche Programmzähler berücksichtigt werden müssen. Zudem ist der zuvor beschriebene Ansatz aufgrund der Verwendung der zusätzlichen Schattenregister relativ aufwendig.

Der vorliegenden Erfindung liegt daher die Aufgabe zugrunde, ein Prozessorsystem mit Interruptverarbeitung vorzuschlagen, bei dem ein durch einen entsprechenden Interrupt hervorgerufener Wechsel des Programmablaufs mit geringerem Aufwand durchgeführt werden kann.

Diese Aufgabe wird erfindungsgemäß durch ein Prozessorsystem mit den Merkmalen des Anspruches 1 gelöst. Die Unteransprüche definieren vorteilhafte und bevorzugte Ausführungsformen der vorliegenden Erfindung.

5

Erfindungsgemäß sind für unterschiedliche Programmabläufe ("Tasks") unterschiedliche Speichermittel vorgesehen, wobei die Speichermittel insbesondere einen Speicher für den Programmzähler des jeweiligen Programmablaufs, Register zum
10 Zwischenspeichern von Daten während der Ausführung des jeweiligen Programmablaufs und Zustandsregister zum Speichern von Zustandsinformationen, wie beispielsweise Carry- oder Zero-Flags, umfassen können. Soll bei Auftreten eines bestimmten Interrupt-Ereignisses von einem Programmablauf auf
15 einen anderen Programmablauf umgeschaltet werden, muß hierzu lediglich auf die dem neuen Programmablauf zugeordneten Speichermittel umgeschaltet werden. Auf diese Weise wird durch das Vorsehen getrennter Hardwarebereiche für unterschiedliche Programmabläufe ein hardwaregestützter
20 Multitasking-Betrieb des Prozessorsystems ermöglicht, ohne daß beim Umschalten auf einen neuen Programmablauf eine Vielzahl von Sicherungsvorgängen durchgeführt werden müssen.

Die Verarbeitung der unterschiedlichen Interrupts erfolgt
25 vorzugsweise prioritätsgesteuert, wobei sowohl den entsprechenden Interrupt-Ereignissen als auch den in dem Prozessorsystem ablaufenden Programmabläufen individuelle Prioritäten zugewiesen sind. Vorzugsweise ist ein bestimmter Prioritätsgrad definiert, der zu einem Programmablaufwechsel
30 führen kann. Nur Ereignisse mit diesem bestimmten Prioritätsgrad können einen Programmablaufwechsel hervorrufen. Alle anderen Ereignisse haben lediglich zur Folge, daß eine Startadresse für die entsprechende Interruptroutine erzeugt wird.

35

Die vorliegende Erfindung eignet sich beispielsweise zum Einsatz in Telekommunikations-Endgeräten, wie z.B. Modems

oder Linecards, wobei dies insbesondere auf xDSL-Endgeräte zutrifft.

Die Erfindung wird nachfolgend näher unter Bezugnahme auf die
5 beigefügte Zeichnung anhand eines bevorzugten Ausführungsbeispiels erläutert.

Fig. 1 zeigt ein vereinfachtes Blockschaltbild eines Prozessorsystems gemäß einem bevorzugten Ausführungsbeispiel
10 der vorliegenden Erfindung,

Fig. 2 zeigt eine mögliche Realisierung einer in Fig. 1 gezeigten Programmsteuerungseinheit,

15 Fig. 3 zeigt eine mögliche Realisierung einer in Fig. 2 gezeigten Ereignisauswertungseinheit, und

Fig. 4 zeigt eine mögliche Realisierung einer in Fig. 3 gezeigten Überwachungseinheit.

20

Das in Fig. 1 gezeigte Prozessorsystem umfaßt eine
Prozessoreinheit 1, die u.a. eine Programmsteuerung 12 sowie
einen Befehlsdecoder 5, eine sogenannte "Branch Control"-
Einheit ("Branch Control Unit", BCU) 6 und mindestens eine
25 Ausführungseinheit ("Algorithmic Logic Unit", ALU) 7 umfaßt.
Darüber hinaus ist für jeden auf dem Prozessorsystem
ausführbaren Programmablauf ("Task") eine individuelle oder
programmablauf-spezifische Hardwareumgebung 8 vorgesehen, die
jeweils einen Stapelspeicher 9, eine Registerbank mit einem
30 oder mehreren Registern 10 und ein oder mehrere
Zustandsregister 11 umfaßt. In dem Stapelspeicher 9 ist
jeweils der augenblickliche Wert des Programmzählers
("Program Counter", PC) des entsprechenden Programmablaufs
gespeichert. Die Register 10 dienen während der Abarbeitung
35 des entsprechenden Programmablaufs zum Zwischenspeichern von
Daten. Die Zustandsregister 11 sind hingegen zum Speichern
von Zustandsinformationen oder Zustandsflags, wie

beispielsweise von Zero- oder Carry-Flags, vorgesehen, so daß der Inhalt der Zustandsregister jeweils den augenblicklichen Zustand des entsprechenden Programmablaufs bezeichnet.

- 5 Die Programmsteuerung 12 überwacht mehrere Interrupt-Ereignisse EV. Bei Auftreten eines bestimmten Interrupt-Ereignisses wird entweder eine entsprechende Interruptroutine gestartet oder ein Programmablaufwechsel durchgeführt. Von der Programmsteuerung 12 wird somit der jeweils aktuelle
10 Programmablauf sowie der auszuführende Befehl innerhalb dieses Programmablaufs und damit der Wert des Programmzählers dieses Programmablaufs festgelegt und entsprechende Informationen ausgegeben.
- 15 Mit Hilfe dieses von der Programmsteuerung 12 ausgegebenen Werts des Programmzählers PC kann auf einen Programmspeicher 2 zugegriffen werden und ein entsprechender Befehl INST ausgelesen werden. Dieser Befehl INST wird von dem Befehlsdecoder 5 decodiert und zur Ausführung der BCU-Einheit
20 6 und der Ausführungseinheit 7 zugeführt, wobei diesen beiden zuletzt genannten Einheiten auch die Nummer des abzuarbeitenden Programmablaufs bekannt ist. Die BCU-Einheit 6 stellt zusammen mit dem Befehlsdecoder 5 sicher, daß stets auf die korrekte Hardwareumgebung 8, d.h. die tatsächlich für
25 den abzuarbeitenden Programmablauf vorgesehene Hardwareumgebung, zugegriffen wird. Dabei ist die BCU-Einheit 6 für eine hardwaremäßige Umschaltung zwischen den Stapelspeichern 9, die für die einzelnen Programmzähler PC vorgesehen sind, verantwortlich, während auf die Registern 10
30 und die Zustandsregistern 11 sowohl mit Hilfe einer hardwaremäßigen Umschaltung durch die BCU-Einheit 6 als auch softwaremäßig mit Hilfe eines internen Befehls durch den Befehlsdecoder 5 zugegriffen wird. Die Ausführungseinheit 7 führt schließlich unter Verwendung der entsprechenden
35 Hardwareumgebung 8 den Befehl INST aus, wobei hierzu die Adressierung eines Datenspeichers 3 erforderlich sein kann, an dem ein Adreßsignal ADR angelegt und mit Hilfe eines

Steuersignals R/W festgelegt wird, ob Daten DATA aus dem Datenspeicher 3 ausgelesen oder in den Datenspeicher 3 geschrieben werden sollen. Ebenso kann infolge der Befehlsausführung die Adressierung eines oder mehrerer Ein-
5 /Ausgabeports 4 notwendig sein, an die ein Adreßsignal ADR anzulegen ist, um Daten DATA aus einem Ausgabeport 4 auslesen bzw. in einen Eingabeport 4 schreiben zu können.

in Fig. 2 ist eine mögliche Realisierung für die
10 Programmsteuerung 12 dargestellt. Eine Auswertungseinheit 13 überwacht das Auftreten der einzelnen Interrupt-Ereignisse EV und erzeugt davon abhängig ein Interrupt-Anforderungssignal IRQ mit einer Startadresse VEC für die entsprechende Interruptroutine oder ein Umschaltsignal SWITCH zur
15 Einleitung eines Programmablaufwechsels mit einer Information TASK über denjenigen Programmablauf, auf den umgeschaltet werden soll. Eine Steuereinheit 21 empfängt diese Informationen von der Auswertungseinheit 13 und veranlaßt davon abhängig, wie bereits beschrieben worden ist, die
20 Verwendung der jeweils korrekten Hardwareumgebung 8. Zu diesem Zweck gibt die Steuereinheit 21 einerseits eine Information CTASK über den augenblicklich auszuführenden Programmablauf an den Befehlsdecoder 5 und die BCU-Einheit 6 sowie den diesem Programmablauf CTASK entsprechenden
25 Programmzähler PC zur Adressierung des Programmspeichers 2 aus.

Die Interruptverarbeitung erfolgt prioritätsgesteuert. Zu diesem Zweck ist jedem Programmablauf und jedem Interrupt-
30 Ereignis EV eine individuelle Priorität zugewiesen. Die Priorität PRIO des augenblicklichen Programmablaufs wird der Auswertungseinheit 13 von der Steuereinheit 21 mitgeteilt. Grundsätzlich kann ein Programmablauf von einem Interrupt nur dann unterbrochen werden, wenn das entsprechende Interrupt-
35 Ereignis eine höhere Priorität als der Programmablauf besitzt. Demzufolge muß die Priorität eines Programmablaufs durch den Programmablauf selbst oder durch einen anderen

- Programmablauf heruntergesetzt werden, falls eine Unterbrechung des entsprechenden Programmablaufs auch durch Interrupt-Ereignisse mit einem geringeren Prioritätsgrad möglich sein soll. Ein Programmablauf, der den höchsten
- 5 Prioritätsgrad besitzt, kann nur durch Interrupt-Ereignisse unterbrochen werden, die ebenfalls den höchsten Prioritätsgrad besitzen. Dies kann beispielsweise auf einen "Reset"-Interrupt zutreffen.
- 10 Innerhalb der verschiedenen Prioritätsgrade ist ein bestimmter Prioritätsgrad definiert, der ausschließlich zu einem Programmablaufwechsel führt. D.h. nur Interrupt-Ereignisse mit diesem Prioritätsgrad können einen Wechsel zwischen den einzelnen Programmabläufen hervorrufen, so daß
- 15 nur bei Auftreten dieser Interrupt-Ereignisse die in Fig. 2 gezeigten Informationen SWITCH und TASK erzeugt und der Steuereinheit 21 zugeführt werden. Alle anderen Interrupt-Ereignisse führen lediglich dazu, daß von der
- 20 Auswertungseinheit 13 die Interruptanforderung IRQ mit einem Vektor VEC, der die Sprung- oder Startadresse für die auszuführende Interruptroutine enthält, ausgegeben wird. Der Steuereinheit 21 werden somit entweder die Signale SWITCH und TASK oder die Signale IRQ und VEC zugeführt. Im ersten Fall erfolgt der Aufruf einer entsprechenden Interruptroutine,
- 25 während im zweiten Fall ein Programmablaufwechsel stattfindet. Das "IRQ"-Signal und das "SWITCH"-Signal können nicht gleichzeitig auftreten.

- In Fig. 3 ist eine mögliche Realisierung für die in Fig. 2
- 30 gezeigte Auswertungseinheit 13 dargestellt, wobei davon ausgegangen wird, daß drei unterschiedlichen Prioritäten definiert sind und die Priorität "2" die exklusiv für einen Programmablaufwechsel vorgesehene Priorität ist. Für jede Gruppe von Interrupt-Ereignissen EV_1...EV_3, die dieselbe
- 35 Priorität besitzen, ist eine eigene Überwachungseinheit 14_1...14_3 vorgesehen, welche ausschließlich das Auftreten der ihr zugewiesenen Interrupt-Ereignisse sowie die Priorität

PRI0 des augenblicklich abgearbeiteten Programmablaufs überwacht.

Sobald die Überwachungseinheit 14_1 oder 14_3 das Auftreten
5 eines ihr zugeordneten Interrupt-Ereignisses EV_1 oder EV_3
erkannt hat, wird ein Interrupt-Anforderungssignal IRQ
erzeugt, und eine nachgeschaltete Einheit 15_1 bzw. 15_3
generiert abhängig von dem aufgetretenen Interrupt-Ereignis
einen entsprechenden Vektor VEC, der die gewünschte
10 Startadresse für die auszuführende Interruptroutine
bezeichnet. Die somit erzeugten Signale werden Multiplexern
16 bzw. 17 zugeführt, wobei die Multiplexer 16 und 17
abhängig von dem Prioritätssteuersignal PRI0 geschaltet
werden und davon abhängig lediglich einen der beiden
15 Eingangssignale an ihren Ausgang durchschalten und der in
Fig. 2 gezeigten Steuereinheit 21 zuführen.

Die Überwachungseinheit 14_2 ist ausschließlich zum
Überwachen derjenigen Interrupt-Ereignisse EV_2 vorgesehen,
20 welche die Priorität "2" besitzen und somit per Definition
einen Wechsel des Programmablaufs hervorrufen können. Sobald
die Überwachungseinheit 14_2 das Auftreten eines ihr
zugeordneten Interrupt-Ereignisses EV_2 erkannt hat, wird das
Umschaltsignal SWITCH erzeugt, welches die Steuereinheit 21
25 anweist, einen Programmablaufwechsel vorzunehmen, und eine
nachgeschaltete Einheit 15_2 erzeugt abhängig von dem
aufgetretenen Interrupt-Ereignis in Form der Information TASK
die Nummer desjenigen Programmablaufs, auf den umgeschaltet
werden soll.

30 Innerhalb jeder Überwachungseinheit 14_1...14_3 können
Unterprioritäten definiert sein, wenn die entsprechende
Überwachungseinheit ihre Eingänge hinsichtlich des Auftretens
der entsprechenden Interrupt-Ereignisse stets in derselben
35 Reihenfolge, beispielsweise stets von oben nach unten,
abtastet.

In Fig. 4 ist eine mögliche Realisierung für die in Fig. 3 gezeigte Überwachungseinheit 14_1 und die nachgeschaltete Einheit 15_1 dargestellt.

5 Die Überwachungseinheit 14_1 umfaßt einen Eingangsspeicher 19, in dem bei Auftreten eines der Interrupt-Ereignisse EV_1 ein entsprechendes Bit gesetzt wird. Eine Bitüberwachungseinheit 20 überwacht die einzelnen Bits des Eingangsspeichers 19 in Übereinstimmung mit einem
10 vorgegebenen Takt CLK. Sobald die Bitüberwachungseinheit 20 in dem Eingangsspeicher 19 ein gesetztes Bit erkannt hat, wird das Interrupt-Anforderungssignal IRQ erzeugt und abhängig von der Position des in dem Eingangsspeicher 19 gesetzten Bits der Vektor VEC mit der Start- oder
15 Sprungadresse für die gewünschte Interrupt-Routine generiert. Zu diesem Zweck wird die Position des in dem Eingangsspeicher 19 gesetzten Bits von der Einheit 15_1 auf die entsprechende Startadresse abgebildet, wobei dies wie in Fig. 4 gezeigt dadurch erfolgt, daß zunächst eine (einer Multiplikation mit dem Faktor 2^n) entsprechende Bitverschiebeoperation um n Bits
20 und anschließend eine Addition mit einem bestimmten Offsetwert OFF durchgeführt wird. Anschließend wird mit Hilfe eines Löschsymbols CLEAR das in dem Eingangsspeicher 19 gesetzte Bit wieder gelöscht.

25 In der Bitüberwachungseinheit 20 wird auch die Priorität PRIO des augenblicklichen Programmablaufs überwacht. Die Erzeugung des Interrupt-Anforderungssignals IRQ und des Vektors VEC mit der Startadresse der entsprechenden Interrupt-Routine wird
30 von der Bitüberwachungseinheit 20 nur freigegeben, wenn die Priorität PRIO des augenblicklichen Programmablaufs kleiner als die Priorität der entsprechenden Interrupt-Ereignisse ist.

In Fig. 4 ist zudem die Möglichkeit dargestellt, daß einzelne
35 Interrupt-Ereignisse mit Hilfe einer Ausblendungs- oder Maskierungsschaltung 18 maskiert, d.h. für die Bitüberwachungseinheit 20 unsichtbar gemacht, werden können.

10

Im vorliegenden Beispiel ist diese Ausblendungsschaltung 18 in Form von AND-Gattern ausgestaltet, die mit Hilfe eines entsprechenden Signals MASK programmiert werden können. Vorzugsweise wird vor jeder Überwachungseinheit 14_1...14_3
5 eine derartige individuell programmierbare Ausblendungsschaltung 18 eingesetzt.

Die Überwachungseinheit 14_3 und die entsprechende Einheit 15_3 können identisch zu Fig. 4 aufgebaut sein. Ebenso kann
10 die Überwachungseinheit 14_2 identisch zu Fig. 4 aufgebaut sein. Der Aufbau der Einheit 15_2 unterscheidet sich hingegen gegenüber Fig. 4 dadurch, daß von der Einheit 15_2 die Position eines in dem entsprechenden Eingangsspeicher 19 gesetzten Bits nicht auf eine Adresse, sondern auf eine
15 Programmablaufnummer abgebildet wird.

Patentansprüche

1. Multitasking-Prozessorsystem,
mit einer Programmsteuerung (12), die eine Auswerteeinheit
5 (13) zum Überwachen des Auftretens von Interrupt-Ereignissen
(EV) und
eine Steuereinheit (21) aufweist, die zum Steuern der
Ausführung einer Interruptroutine vorgesehen ist, wenn von
der Auswerteeinheit (13) das Auftreten eines Interrupt-
10 Ereignisses (EV) erkannt worden ist,
wobei in dem Multitasking-Prozessorsystem für verschiedene
Programmabläufe unterschiedliche Speicherschaltungen (8)
vorgesehen sind, und
die Steuereinheit (21) derart ausgestaltet ist, dass sie bei
15 Auftreten eines bestimmten Interrupt-Ereignisses (EV),
welches einen Wechsel von einem augenblicklichen
Programmablauf zu einem neuen Programmablauf hervorruft und
welches einen bestimmten vorgegebenen Prioritätsgrad
aufweist, zur Ausführung des neuen Programmablaufs ein
20 Umschalten von der dem bisherigen Programmablauf zugeordneten
Speicherschaltung (8) auf die dem neuen Programmablauf
zugeordneten Speicherschaltung (8) veranlasst,
wobei die Speicherschaltung (8) jeweils einen Speicher (9)
für einen Programmzähler des zugehörigen Programmablaufs,
25 mindestens ein Register (10) zum Zwischenspeichern von Daten,
die bei der Ausführung des zugehörigen Programmablaufs
anfallen, und
mindestens ein Zustandsregister (11) zum Speichern von
Zustandsinformationen des zugehörigen Programmablaufs
30 aufweist, wobei den einzelnen Interrupt-Ereignissen
individuelle Prioritäten zugeordnet sind und mindestens eine
bestimmte Priorität für einen Wechsel zu einem neuen
Programmablauf vorgesehen ist, und
wobei die Auswerteeinheit (13) bei Erkennen des Auftretens
35 eines Interrupt-Ereignisses (EV) mit der bestimmten Priorität
eine entsprechende Aufforderung (SWITCH) für einen Wechsel

von dem augenblicklichen Programmablauf zu dem entsprechenden neuen Programmablauf an die Steuereinheit (21) weiterleitet.

2. Prozessorsystem nach Anspruch 1,

- 5 d a d u r c h g e k e n n z e i c h n e t,
dass den einzelnen, von dem Prozessorsystem ausführbaren Programmabläufen individuelle Prioritäten zugeordnet sind, dass den einzelnen Interrupt-Ereignissen individuelle Prioritäten zugeordnet sind, und
- 10 dass die Auswerteeinheit (13) derart ausgestaltet ist, dass sie bei Erkennen des Auftretens eines Interrupt-Ereignisses eine entsprechende Aufforderung (IRQ) zur Ausführung des entsprechenden Interrupts an die Steuereinheit (21) nur dann weiterleitet, falls die Priorität des entsprechenden
- 15 Interrupt-Ereignisses höher als die Priorität (PRIO) des in dem Prozessorsystem augenblicklich ablaufenden Programmablaufs ist.

3. Prozessorsystem nach Anspruch 2,

- 20 d a d u r c h g e k e n n z e i c h n e t,
dass die Auswerteeinheit (13) derart ausgestaltet ist, dass sie bei Erkennen des Auftretens eines Interrupt-Ereignisses mit der bestimmten Priorität zusammen mit der Aufforderung (SWITCH) für den Wechsel von dem augenblicklichen
- 25 Programmablauf zu dem entsprechenden neuen Programmablauf auch eine den neuen Programmablauf spezifizierende Information (TASK) an die Steuereinheit (21) weiterleitet.

4. Prozessorsystem nach Anspruch 2 oder 3,

- 30 d a d u r c h g e k e n n z e i c h n e t,
dass die Auswerteeinheit (13) derart ausgestaltet ist, dass sie bei Erkennen des Auftretens eines Interrupt-Ereignisses eine entsprechende Aufforderung (IRQ) zur Ausführung des entsprechenden Interrupts zusammen mit einer
- 35 entsprechenden Interrupt-Routine spezifizierenden Adresse (VEC) an die Steuereinheit (21) nur dann weiterleitet, falls die Priorität des entsprechenden Interrupt-Ereignisses höher

13

als die Priorität (PRIO) des in dem Prozessorsystem-
augenblicklich ablaufenden Programmablaufs ist und die
Priorität des Interrupt-Ereignisses nicht der bestimmten
Priorität entspricht.

5

5. Prozessorsystem nach einem der vorhergehenden Ansprüche,
dadurch gekennzeichnet,
dass die Auswerteeinheit (13) für jede mögliche Priorität der
Interrupt-Ereignisse (EV) eine separate

10 Überwachungseinrichtung (14) umfasst, wobei jede
Überwachungseinrichtung (14) das Auftreten der Interrupt-
Ereignisse (EV) mit der jeweils entsprechenden Priorität
überwacht.

15 6. Prozessorsystem nach einem der vorhergehenden Ansprüche,
dadurch gekennzeichnet,
dass der Auswerteeinheit (13) eine programmierbare
Maskierungsschaltung (18) vorgeschaltet ist, die abhängig von
ihrer Programmierung das Auftreten entsprechender Interrupt-
20 Ereignisse (EV) für die Auswerteeinheit (13) maskiert.

7. Prozessorsystem nach Anspruch 5 und 6,
dadurch gekennzeichnet,
dass die Auswerteeinheit (13) für jede
25 Überwachungseinrichtung (14) eine dieser
Überwachungseinrichtung (14) vorgeschaltete separate
Maskierungsschaltung (18) aufweist.

8. Verwendung eines Prozessorsystems nach einem der
30 vorhergehenden Ansprüche in einem Telekommunikations-
Endgerät, insbesondere in einem xDSL-Telekommunikations-
Endgerät.

35

FIG 1

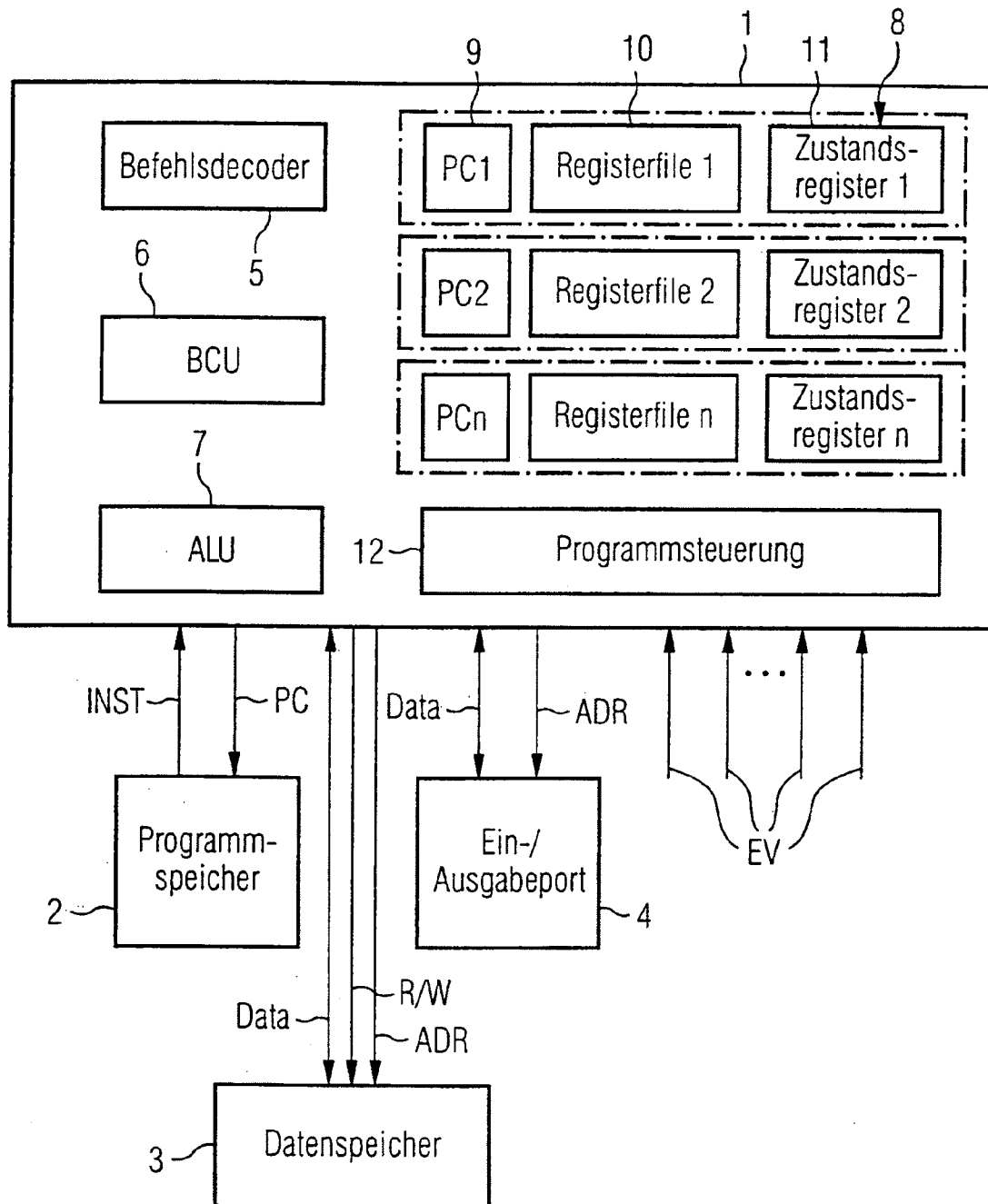


FIG 2

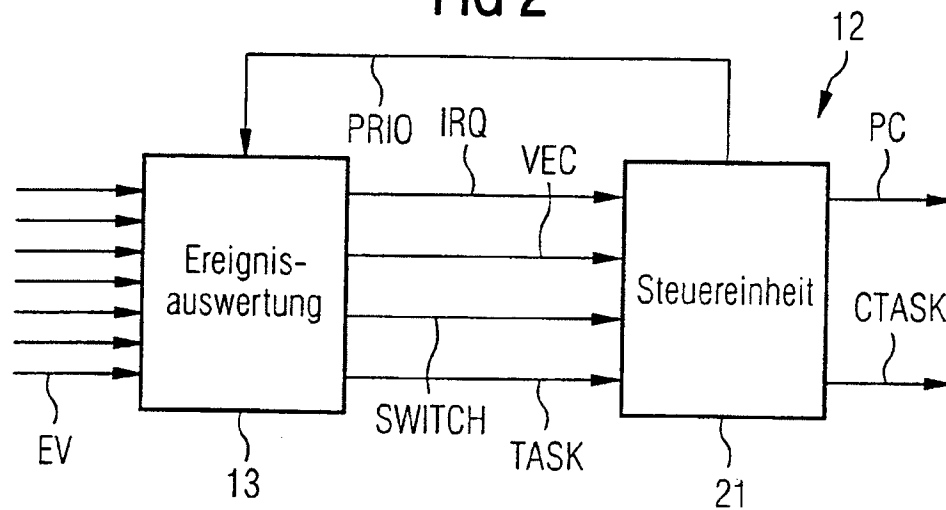


FIG 3

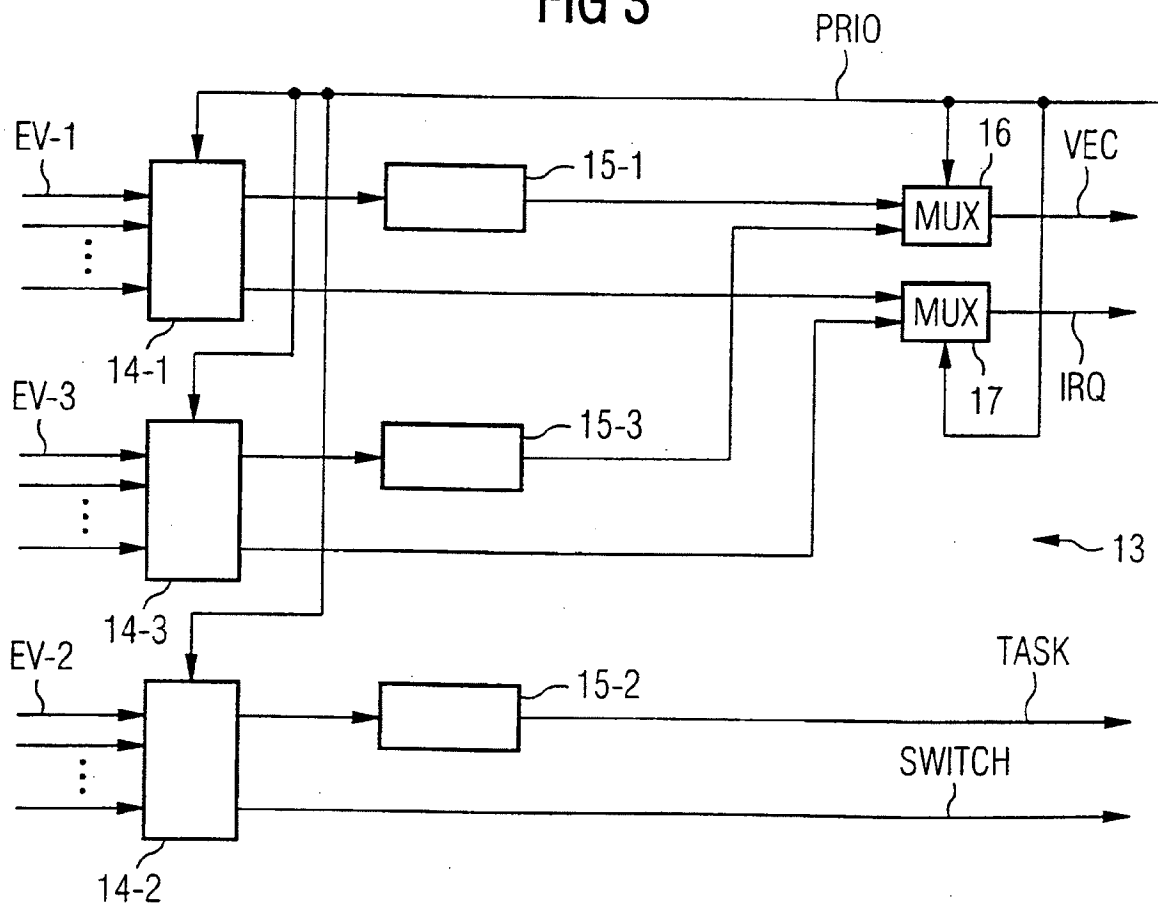


FIG 4

